Bike Sharing Demand Prediction

by
Shaik Ahmad Basha
Data science trainee,
AlmaBetter, Bangalore.

Abstract:

As a convenient, economical, and ecofriendly travel mode, bike-sharing greatly improved urban mobility. However, it is often very difficult to achieve a balanced utilization of shared bikes due to the asymmetric user demand distribution and the insufficient numbers of shared bikes, docks, or parking areas. If we can predict the short-run bike-sharing demand, it will help operating agencies rebalance bike-sharing systems in a timely and efficient way.

1.Problem Statement

Currently Rental bikes are introduced in many urban cities for the enhancement of mobility comfort. It is important to make the rental bike available and accessible to the public at the right time as it lessens the waiting time. Eventually, providing the city with a stable supply of rental bikes becomes a major concern. The crucial part is the prediction of bike count required at each hour for the stable supply of rental bikes.

Our main objective behind this project is to explore and analyze the data to discover the key understandings. And to predict the count of bikes required at each hour by using regression models

2.Data Description

The dataset contains weather information (Temperature, Humidity, Windspeed, Visibility, Dewpoint, Solar radiation, Snowfall, Rainfall), the number of bikes rented per hour and date information.

The dataset has 8670 rows and 14 columns. Those Features are

- Date : year-month-day
- Rented Bike count - Count of bikes rented at each hour
- Hour - Hour of he day
- Temperature-Temperature in Celsius
- Humidity - %
- Windspeed - m/s
- Visibility - 10m
- Dew point temperature - Celsius
- Solar radiation - MJ/m2
- Rainfall - mm
- Snowfall - cm
- Seasons - Winter, Spring, Summer, Autumn
- Holiday - Holiday/No holiday
- Functional Day - NoFunc(Non Functional Hours), Fun(Functional hours)

3. Introduction:

Currently Rental bikes are introduced in many urban cities for the enhancement of mobility comfort. It is important to make the rental bike available and accessible to the public at the right time as it lessens the waiting time. Eventually, providing the city with a stable supply of rental bikes becomes a major concern.

So Exploratory data analysis on these datasets will give some understandings.

Prediction of bike count required at each hour is very important to be successful.

4. Steps Involved

Data Understanding

After the loading and collecting the dataset, understanding the data is very important. I Understand the various features of the dataset.

Data Cleaning

Cleaning and manipulating the data is very important. Only the cleaned data can be fitted into any machine learning models.

Our dataset contains no null values and duplicated values.

I converted the 'Date' feature to datetime feature and 'Hour' feature to Categorical feature.

A new column 'Day' is created which contains day name. And also weekend column is created which contains 0(is not weekend) and 1(is weekend)

Exploratory Data Analysis

After handling with null values, duplicated values, and manipulating the data, we can deep dive into Exploratory Data Analysis. Here I started with dividing the features into numerical, categorical features and datetime features. And then observed the distribution of numerical features and how these features are related with dependent feature. I observed that independent features are not linearly related to the dependent feature. And then I observed relationship between categorical features and dependent feature.

Removing Multicollinearity

Multicollinearity occurs when two or more independent variables are highly correlated with one another in a regression model. This means that an independent variable can be predicted from another independent variable in a regression model.

By visualizing the correlation heatmap, I observed that there is a multicollinearity exists in the data. The two features namely 'Temperature' and 'Dew Point Temperature' are highly correlated with each other.

So I removed 'Dew Point Temperature' because it is less correlated with dependent feature than 'Temperature'

Removing Outliers

An outlier is a data point that does not fit with the rest of the data. An outlier can be higher or lower than expected, or displaced more to the right or left than expected. Outliers can effect regression lines, making the regression lines less accurate in predicting other data.

So identifying the outliers and removing them is very important before fitting the data into model.

Here there are various outliers present in the data. So at first I created a 'for' loop for all numerical features which append the outlier index value (determined using Z-Score) into a variable, and then I simply dropped them from the data.

Encoding of categorical features

Machine learning models can only work with numerical values. For this reason, it is necessary to transform the categorical values of the relevant features into numerical ones. This process is called feature encoding.

I used pd.get_dummies() function to convert categorical features into numerical features.

Feature Transformation:

Transformation of the skewed variables may also help correct the distribution of the variables. These could be logarithmic, square root, or square transformations. In our dataset Dependent variable('Rented Bike Count)having a moderate right skewed, to apply linear regression dependent features have to follow the normal distribution. Therefore, we use square root transformation on top of it.

Feature Scaling

Machine learning is like making a mixed fruit juice. If we want to get the best-mixed juice, we need to mix all fruit not by their size but based on their right proportion. We just need to remember apple and strawberry are not the same unless we make them similar in some context to compare their attribute. Similarly, in many machine learning algorithms, to bring all features in the same standing, we need to do scaling so that one significant number doesn't impact the model just because of their large magnitude.

Feature scaling in machine learning is one of the most critical steps during the pre-processing of data before creating a machine learning model. Scaling can make a difference between a weak machine learning model and a better one.

So I scaled the features by using MinMaxScaler.

Evaluation Metrics

Evaluation metrics are a measure of how good a model performs and how well it approximates the relationship. Let us look at MAE, MSE, R-squared, Adjusted Rsquared, and RMSE.

R-squared:

R-square is a comparison of residual sum of squares (SSres) with total sum of squares(SStot).

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}}$$

Mean Absolute Error (MAE):

This is simply the average of the absolute difference between the target value and the value predicted by the model.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right|$$

Mean Squared Error (MSE):

The most common metric for regression tasks is MSE. It has a convex shape. It is the average of the squared difference between the predicted and actual value.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2$$

Root Mean Squared Error (RMSE):
This is the square root of the average of the squared difference of the predicted and actual value.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

Adjusted R-Squared:
The main difference between adjusted Rsquared and R-square is that Rsquared describes the amount of variance of the dependent variable represented by every single independent variable, while adjusted R-squared measures variation explained by only the independent variables that actually affect the dependent variable.

$$R^2_{adjusted} = \left[\frac{(1-R^2)(n-1)}{n-k-1}\right]$$

Fitting the data into different models

1. **Linear Regression**
2. **Lasso Regression**
3. **Ridge regression**
4. **Decision Tree Regressor**
5. **Random Forest regressor**
6. **Gradient Boosting Regressor**
7. **XGBoost Regressor**

1. Linear Regression

It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, age, product price, etc. Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called linear regression.

Cost Function (J):
Cost function(J) of Linear Regression is the Root Mean Squared Error (RMSE) between predicted y value (pred) and true y value (y)

$$J = \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

2. Lasso Regression

Lasso regression is a type of linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, like the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of multicollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination. Lasso regression performs L1 regularization, which adds a penalty equal to the absolute value of the magnitude of coefficients. This type of regularization can result in sparse models with few coefficients; Some coefficients can become zero and eliminated from the model. Larger penalties result in coefficient values closer to zero, which is the ideal for producing simpler models

3. Ridge Regression

Ridge regression is a model tuning method that is used to analyse any data that suffers from multicollinearity. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values being far away from the actual values.

The cost function for ridge regression:

$$Min(||Y - X(theta)||^{\wedge}2 + \lambda||theta||^{\wedge}2)$$

Lambda is the penalty term. λ given here is denoted by an alpha parameter in the ridge function. So, by changing the values of alpha, we are controlling the penalty term. The higher the values of alpha, the bigger is the penalty and therefore the magnitude of coefficients is reduced.

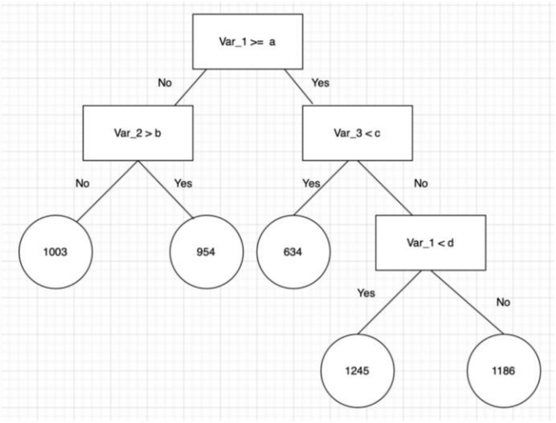
4. Decision Tree Regressor:

Decision Tree is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs, and utility. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables.

The branches/edges represent the result of the node and the nodes have either:

1. Conditions [Decision Nodes]
2. Result [End Nodes]

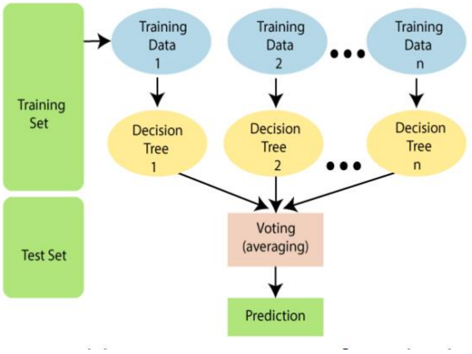
The branches/edges represent the truth/falsity of the statement and take makes a decision based on that in the example below which shows a decision tree that evaluates the smallest of three numbers:



Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values.

5. Random Forest Regressor

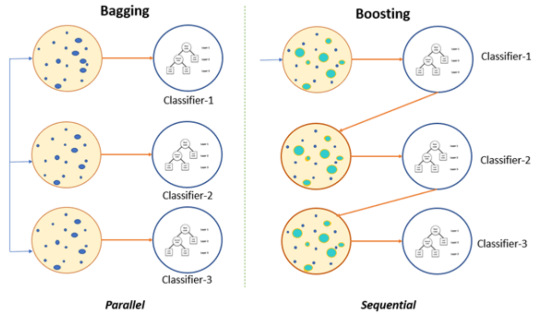
Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.



Ensemble uses two types of methods:

Bagging– It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example, Random Forest.

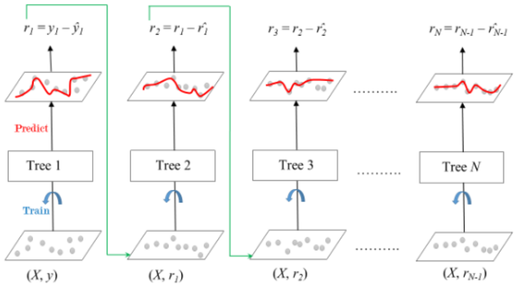
Boosting– It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. For example, ADA BOOST, XG BOOST.



6. Gradient Boosting Regressor:

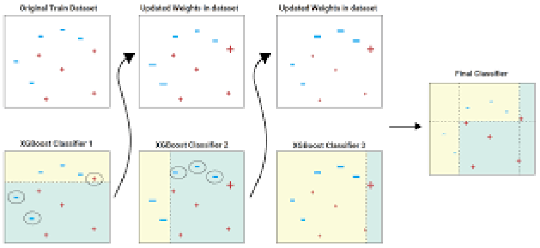
GRADIENT BOOSTING: Gradient Boosting is a popular boosting algorithm. In gradient boosting, each predictor corrects its predecessor's error. There is a technique called the Gradient Boosted Trees whose base learner is CART (Classification and Regression Trees).

The ensemble consists of N trees. Tree1 is trained using the feature matrix X and the labels y. The predictions labelled y1(hat) are used to determine the training set residual errors r1. Tree2 is then trained using the feature matrix X and the residual errors r1 of Tree1 as labels. The predicted results r1(hat) are then used to determine the residual r2. The process is repeated until all the N trees forming the ensemble are trained.



7. XGBoost Regressor:

In this algorithm, decision trees are created in sequential form. Weights play an important role in XGBoost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results. The weight of variables predicted wrong by the tree is increased and these variables are then fed to the second decision tree. These individual classifiers/predictors then ensemble to give a strong and more precise model. XGBoost comes under the boosting ensemble techniques which combines the weakness of primary learners to the next strong and compatible learners.



Conclusion:

Starting with loading the data so far, we have done EDA, outlier treatment, removing multicollinearity, encoding of categorical columns, feature selection and then model building.

After fitting the data into different regression models,

Tree based models performs well than linear models because independent feature are not linearly correlated with dependent feature.

From tree based models, XGBOOST and Gradient Boosting Regressor models gives us the highest score ranging between 85-96%. Functioning day is the most important feature and Winter is the second most for XGBoost Regressor.